05 - Strings in Python

Ex. No. : 5.1 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

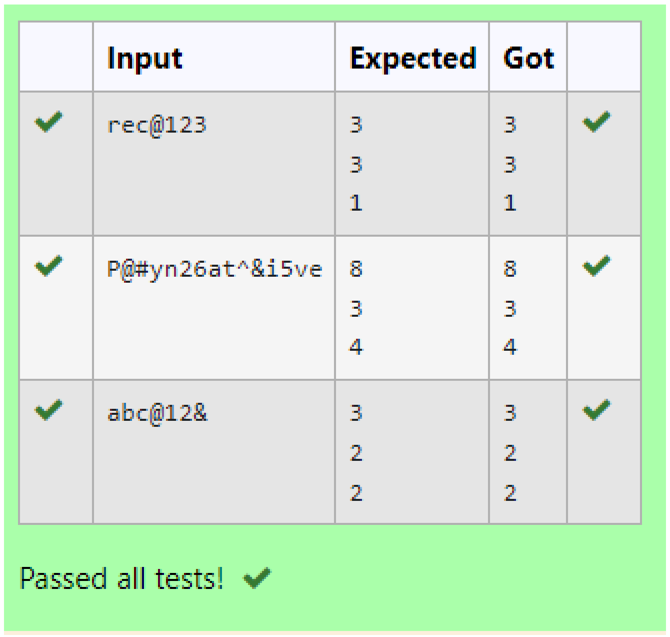
Count Chars

Write a python program to count all letters, digits, and special symbols respectively from a given string

For example:

```
Input  Result
rec@123
3
3
1
```

```
l=0
d=0
s=0
str=input()
for i in str:
    if i.isalpha():
        l+=1
    elif i.isdigit():
        d+=1
    else:
        s+=1
print(l)
print(d)
print(s)
```



	Input	Expected	Got	
✓	`rec@123`	3 3 1	3 3 1	✓
✓	`P@#yn26at^&i5ve`	8 3 4	8 3 4	✓
✓	`abc@12&`	3 2 2	3 2 2	✓

Passed all tests! ✓

Ex. No. : 5.2 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

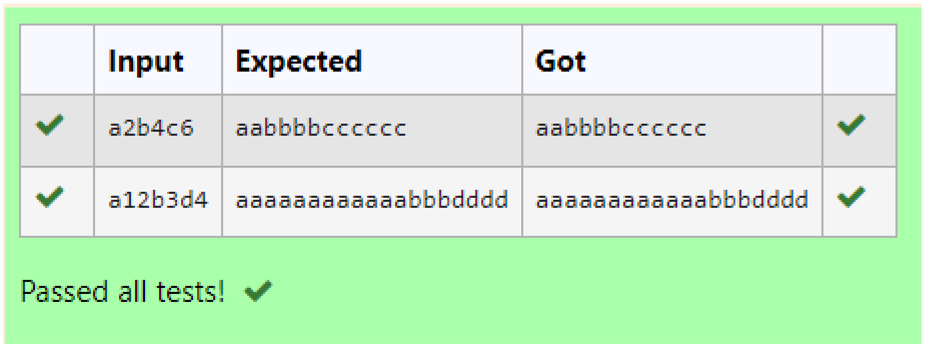
Decompress the String

Assume that the given string has enough memory. Don't use any extra space(IN-PLACE)

Sample Input 1
a2b4c6

Sample Output 1
aabbbbcccccc

```
input_string=input()
result = ""
i = 0
while i < len(input_string):
    char = input_string[i]
    if i + 1 < len(input_string) and input_string[i + 1].isdigit():
        j = i + 1
        while j < len(input_string) and input_string[j].isdigit():
            j += 1
        count = int(input_string[i + 1:j])
        result += char * count
        i = j
    else:
        result += char
        i += 1
print(result)
```



	Input	Expected	Got	
✓	a2b4c6	aabbbbcccccc	aabbbbcccccc	✓
✓	a12b3d4	aaaaaaaaaaaabbbdddd	aaaaaaaaaaaabbbdddd	✓

Passed all tests! ✓

Ex. No. : 5.3 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

First N Common Chars

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

Input Format:

The first line contains S1.
The second line contains S2.
The third line contains N.

Output Format:

The first line contains the N characters present in S1 which are also present in S2.

Boundary Conditions:

2 <= N <= 10
2 <= Length of S1, S2 <= 1000

Example Input/Output 1:

Input:

abcbde
cdefghbb
3

Output:

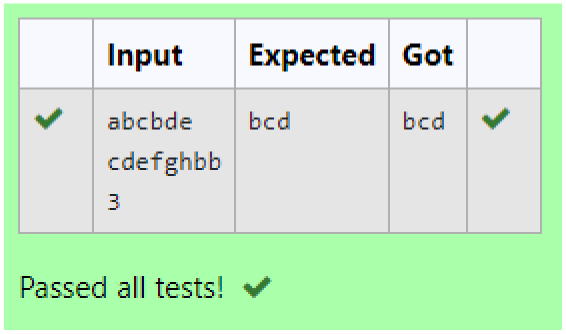
bcd

Note:

b occurs twice in common but must be printed only once.

```
s1=input()

s2=input()

n=int(input())
```

```
common_chars=set(s1)&set(s2)
result=""
for char in s1:
    if char in common_chars:
        result+=char
        common_chars.remove(char)
    if len(result)==n:
        break
print(result[:n])
```



	Input	Expected	Got	
✓	abcbde cdefghbb 3	bcd	bcd	✓

Passed all tests! ✓

Ex. No. : 5.4 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Remove Characters

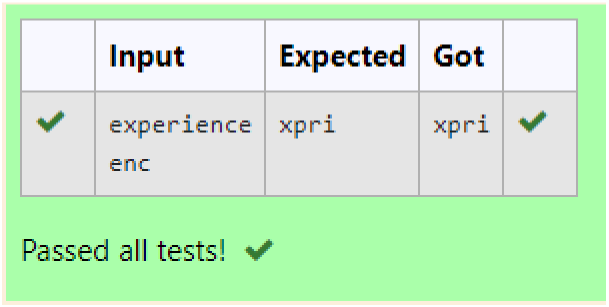
Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints
1<= string length <= 200

Sample Input 1
experience
enc

Sample Output 1
xpri

```
s1=input()
s2=input()
s2_set = set(s2)
result = ""
for char in s1:
    if char not in s2_set:
        result += char
print(result)
```



	Input	Expected	Got	
✓	experience enc	xpri	xpri	✓

Passed all tests! ✓

Ex. No. : 5.5 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

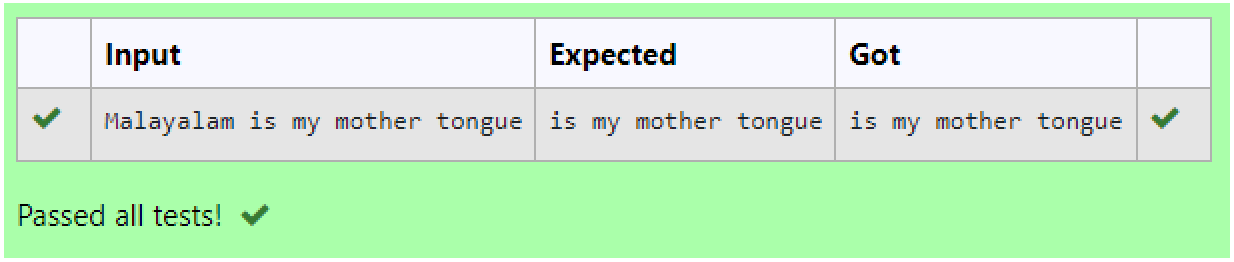
Remove Palindrome Words

String should contain only the words are not palindrome.

Sample Input 1
Malayalam is my mother tongue

Sample Output 1
is my mother tongue

```
def is_palindrome (word):
   return word == word[::-1]
def filter_non_palindromic_words(input_string):
   words = input_string.split()
   non_palindromic_words = [word for word in words if not is_palindrome
(word)]
   return ' '.join(non_palindromic_words)
input_string = input().lower()
output_string = filter_non_palindromic_words (input_string)
print(output_string)
```



	Input	Expected	Got	
✓	Malayalam is my mother tongue	is my mother tongue	is my mother tongue	✓

Passed all tests! ✓

Ex. No. : 5.6 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Return Second World in Uppercase

Write a program that takes as input a string (sentence), and returns its second word in uppercase.

For example:

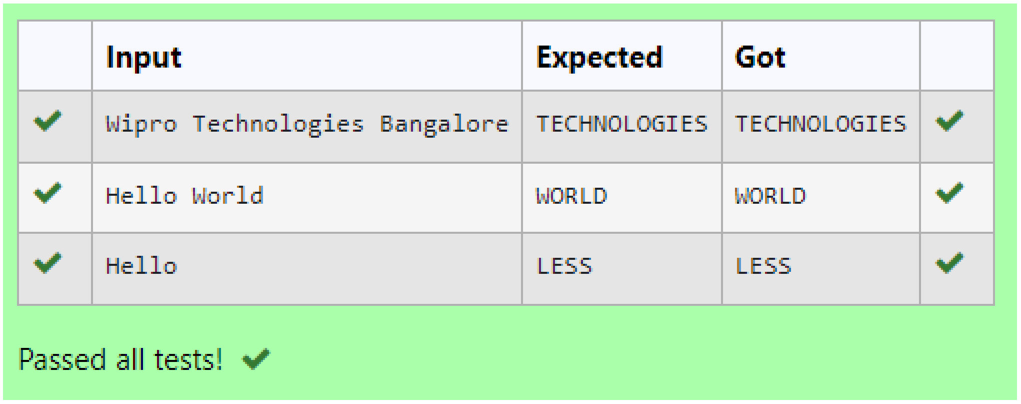
If input is “Wipro Technologies Bangalore” the function should return “TECHNOLOGIES”
If input is “Hello World” the function should return “WORLD”
If input is “Hello” the program should return “LESS”

NOTE 1: If input is a sentence with less than 2 words, the program should return the word “LESS”.
NOTE 2: The result should have no leading or trailing spaces.

For example:

```
Input  Result
Wipro Technologies Bangalore
TECHNOLOGIES
Hello World
WORLD
Hello
LESS
```

```
sentence=input()
words=sentence.split()
if len(words)<2:
  print("LESS")
else:
  print(words[1].upper())
```



	Input	Expected	Got	
✓	`Wipro Technologies Bangalore`	`TECHNOLOGIES`	`TECHNOLOGIES`	✓
✓	`Hello World`	`WORLD`	`WORLD`	✓
✓	`Hello`	`LESS`	`LESS`	✓

Passed all tests! ✓

Ex. No. : 5.7 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Revers String

Reverse a string without affecting special characters. Given a string S, containing special characters and all the alphabets, reverse the string without affecting the positions of the special characters.

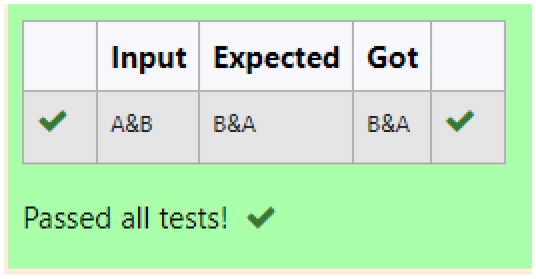
Input:
A&B
Output:
B&A
Explanation: As we ignore '&' and
As we ignore '&' and then reverse, so answer is "B&A".

For example:

Input Result
A&x#
x&A#

```
string=input()

string_list = list(string)

left = 0

right = len(string_list) - 1

while left < right:

    if not string_list[left].isalpha():

        left += 1

        continue

    if not string_list[right].isalpha():

        right -= 1

        continue

    string_list[left], string_list[right] = string_list[right], string_list[left]
```

```
        left += 1
        right -= 1
reversed_string = ''.join(string_list)
print(reversed_string)
```



	Input	Expected	Got	
✓	A&B	B&A	B&A	✓

Passed all tests! ✓

Ex. No. : 5.8 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

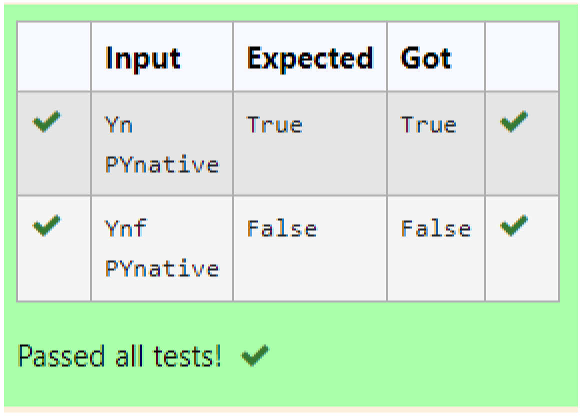
String characters balance Test

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter. If balanced display as "true" ,otherwise "false".

For example:

Input Result
Yn
PYnative
True

```
s1=input()
s2=input()
s1=s1.lower()
s2=s2.lower()
def balance(s1,s2):
    for i in s1:
        if i not in s2:
            return False
    return True
print(balance(s1,s2))
```



	Input	Expected	Got	
✓	Yn PYnative	True	True	✓
✓	Ynf PYnative	False	False	✓

Passed all tests! ✓

Ex. No. : 5.9 **Date: 2/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Unique Names

In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were first entered. For example, if the user enters:

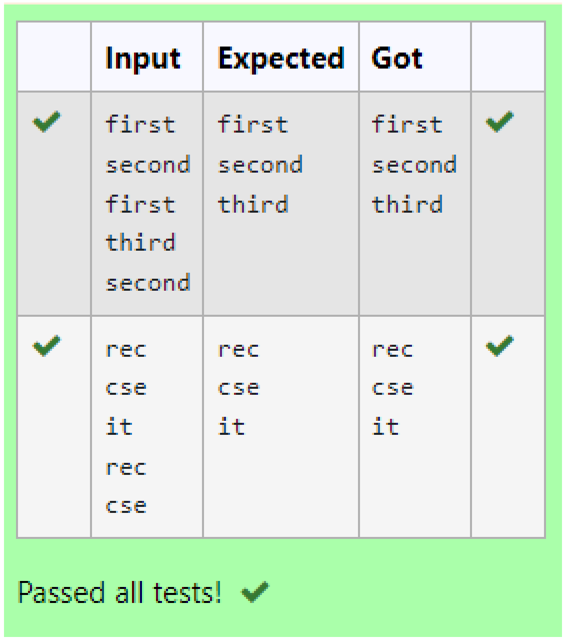
Input:

first
second
first
third
second

then your program should display:

Output:
first
second
third

```
a=set()
for i in iter(input, ""):
    if i not in a:
        a.add(i)
        print(i)
```



	Input	Expected	Got	
✓	first second first third second	first second third	first second third	✓
✓	rec cse it rec cse	rec cse it	rec cse it	✓

Passed all tests! ✓

Ex. No. :	**5.10**	**Date: 2/05/2024**
Register No.:	**231401001**	**Name: Aafrin Fathima N**

Username Domain Extension

Given a string S which is of the format USERNAME@DOMAIN.EXTENSION, the program must print the EXTENSION, DOMAIN, USERNAME in the reverse order.

Input Format:
The first line contains S.

Output Format:
The first line contains EXTENSION.
The second line contains DOMAIN.
The third line contains USERNAME.

Boundary Condition:
1 <= Length of S <= 100

Example Input/Output 1:

Input:

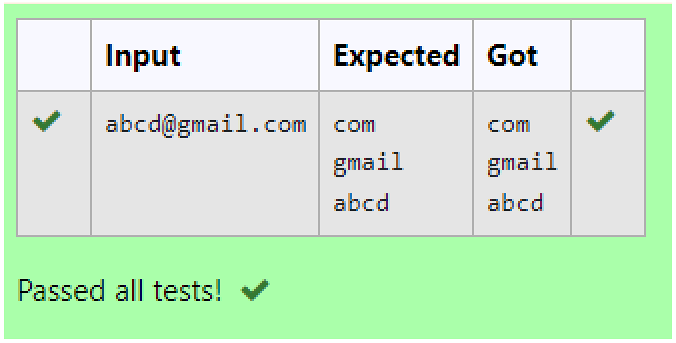
```
vijayakumar.r@rajalakshmi.edu.in
```

Output:

```
edu.in
rajalakshmi
vijayakumar.r
```

```
email = input().strip()

at_index = email.index('@')

dot_index = email.index('.')

username = email[:at_index]

domain = email[at_index+1:dot_index]
```

```
extension = email[dot_index+1:]

print(extension)

print(domain)

print(username)
```



	Input	Expected	Got	
✓	abcd@gmail.com	com gmail abcd	com gmail abcd	✓

Passed all tests! ✓